2

Masking Failures of Multidimensional Sensors

Paul Chew*
Keith Marzullo**

TR 91-1190
February 1991

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

*This author was supported by the Defense Advanced Research Projects Agency (DoD) under ONR Contract N00014-88-K-0591, and by ONR Grant N00014-89-J-1946 and NSF Grant IRI-9006137.
**This author was supported by the Defense Advanced Research Projects Agency (DoD) under NASA Ames grant number NAG 2-593, Contract N00140-87-C-8904, and by grants from IBM and Siemens.

91 5 10 021

Masking Failures of Multidimensional Sensors

Paul Chew* Keith Marzullo†

Cornell University
Department of Computer Science
Ithaca, New York 14853

January 8, 1991

Abstract

When a computer monitors a physical process, the computer uses *sensors* to determine the values of the physical variables that represent the state of the process. A sensor can sometimes fail, however, and in the worst case report a value completely unrelated to the true physical value. The work described in this paper is motivated by a methodology for transforming a process control program that cannot tolerate sensor failure into one that can. In this methodology, a *reliable abstract sensor* is created by combining information from several real sensors that measure the same physical value. To be useful, an abstract sensor must deliver reasonably accurate information at reasonable computational cost.

In this paper, we consider sensors that deliver multidimensional values (e.g., location or velocity in 3 dimensions). Geometric techniques are used to derive upper bounds on abstract sensor accuracy and to develop efficient algorithms for implementing abstract sensors.

1 Introduction

A *process control program* communicates and synchronizes with a physical process. Typically, the program reads values from the physical process

*This author was supported by the Defense Advanced Research Projects Agency (DoD) under ONR Contract N00014-88-K-0591, and by ONR Grant N00014-89-J-1946 and NSF Grant IRI-9006137.

†This author was supported by the Defense Advanced Research Projects Agency (DoD) under NASA Ames grant number NAG 2-593, Contract N00140-87-C-8904, and by grants from IBM and Siemens.

through sensors and writes values through actuators, as shown schematically in Figure 1. This paper is concerned with control programs tolerating failures of continuous-valued sensors.

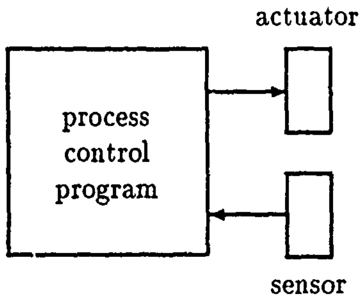

Figure 1: A process-control program

In an earlier paper [6], we presented a methodology for writing process control program that can tolerate faulty sensors:

1. A specification of the control program is written in terms of the state variables of the physical system. For example, the specification of a program controlling a chemical reaction vessel would refer to a variable T whose value is assumed to be the temperature of the vessel.

2. Each physical state variable referenced by the specification is replaced with a reference to an *abstract sensor*. An abstract sensor is a set of values that contains the correct value of the physical variable of interest. Uncertainty in sensor values now becomes an issue, and the specification must be re-examined and possibly changed to accommodate it.

3. The control program is written based on the specification produced by Step 2. This program reads abstract sensors that are assumed to always contain the correct value of the corresponding physical variables.

4. For each abstract sensor referenced by the program written in Step 3, a set of abstract sensors that fail independently are constructed. Each abstract sensor is implemented using a *concrete sensor*, which is a

physical device that "reads" a physical variable, such as a thermometer. This step will require some knowledge of the physical process being controlled as well as the specification of the concrete sensor.

5. A *fault-tolerant averaging function* is used with these replicated abstract sensor values in order to calculate another abstract sensor that is correct even if some of the original sensors are incorrect. The averaging algorithm assumes that no more than f out of the n abstract sensors are incorrect. The relation between n and f depends on the way sensors can fail.

The resulting system will have a structure like that shown in Figure 2.

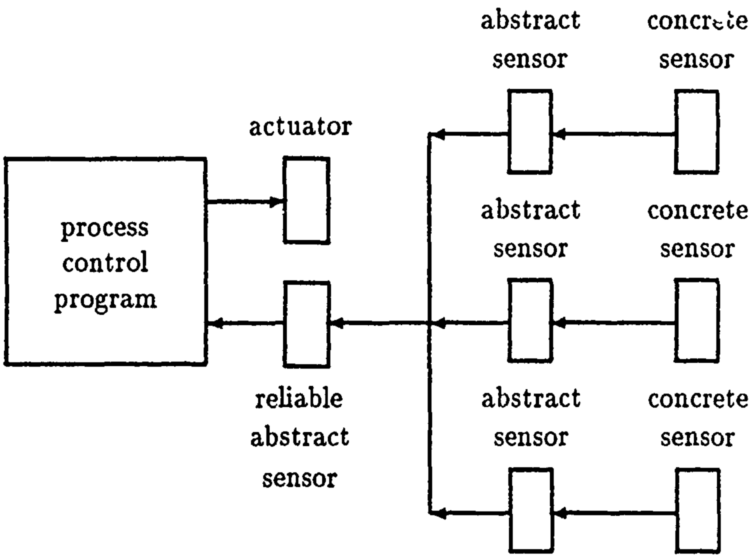

Figure 2: Replicated sensors

Step 5 in the above methodology is an example of masking failures through redundancy [11]. In fact, the fault-tolerant averaging function presented in [6] is a generalization of NMR, or *n-module redundancy*, whereby n independent copies are fed into a majority voter [12]. For both NMR and our averaging function, up to $f = \left\lfloor \frac{n-1}{2} \right\rfloor$ signal failures can be masked.

One limitation of our earlier work is that the fault-tolerant averaging function of [6] is applicable only to sensors that measure a single, independent, real value. An example of a sensor that does not fit this model is one that measures the location of some physical object in 3D space. If such multidimensional sensors are used then a naive approach to masking failures is to consider the x component separately from failures of the y and z components, but doing so limits the accuracy of the resulting value. For example, any sensor found to be faulty by examining the x components should most likely be discarded when considering the y and z components.

In this paper, we extend our fault-tolerant averaging function to multidimensional sensors. We derive the amount of replication necessary to achieve fault masking, which turns out to be a function of the number of possible failures and both the shape and number of dimensions of the sensor measurement. We also discuss efficient algorithms for computing the fault-tolerant average.

One way in which our approach is unusual is that we apply a very weak failure model to sensor failures. This failure model—defining a fault hierarchy and assuming no more than f of n components are faulty—has been applied to several problems in distributed systems such as consensus [8] and reliable broadcast [1]. It has also been incorporated into a methodology for building fault-tolerant distributed programs [10,5]. In contrast to our method of tolerating sensor failure, the more typical approach models the value of a sensor as as a random variable and then convolving several measurements, either from different sensors or the same sensor read at different times [2]. Doing so posits a probability distribution function, which may be too strong an assumption. One of the goals of our research is to understand the applicability of the weaker failure model to continuous-valued signals.

The paper proceeds as follows. In Section 2, we present our failure model for sensors and describe how faults can be masked. Section 3 summarizes the relevant results from [6]. Sections 4 and 5 extends the results of Section 3 to d-dimensional rectangles and d-dimensional circles, respectively. Note that the results on circles actually hold for any class of convex shapes in which the shapes are geometrically similar and share the same orientation (for example, squares aligned with a fixed coordinate system). Section 6 presents discusses bounds for some special cases, and Section 7 summarizes our results

2 System Model

We distinguish between a *concrete sensor*, which is a device that reads a physical state variable and an *abstract sensor* which is a set of possible values for the physical state variable. Abstract sensors are easier to reason about than concrete sensors, in part because there are several different kinds of concrete sensors, each with a different specification. If considered as a whole, the only failure model one can impose on concrete sensors is a probabilistic one. This is not the case for abstract sensors, as discussed below. Further discussion on the implementation of abstract sensors and their use in specifications can be found in [6].

We assume that abstract sensors have the following properties. Let s_i be a sensor of some physical variable $\overline{v}$. A measurement s_i is a continuous set of values that conform to some shape, such as a continuous interval, a rectangle, a sphere, etc. We say that s_i is *correct* if it is not too inaccurate and always includes the value of the actual physical variable. More precisely, for some upper bound acc on the accuracy of s_i,

$$s_i \text{ correct} \overset{\text{def}}{=} \overline{v} \in s_i \ \wedge \ |s_i| \leq acc$$

where $|s_i|$ is the accuracy of s_i. Thus, an abstract sensor can fail in two ways: it can fail to contain the true value or it can report a region so large as to be useless. In this paper, we first assume such large-region sensors could be detected and discarded by preprocessing the abstract sensor data (n and f would have to be adjusted). We relax this assumption in Section 6.

Let s_i and s_j $(i \neq j)$ be the measurements by two abstract sensors for the same physical value $\overline{v}$. If s_i and s_j both contain the correct value, then the measurements s_i and s_j must intersect, and their intersection must contain the (unknown) value $\overline{v}$.

Consider a set $S = \{s_1, s_2, \ldots, s_n\}$ of n independent measurements of the same physical value. If f or less measurements do not contain the correct value, then any set of $n-f$ mutually intersecting measurements may contain the correct value within their intersection, since they each share a common value. Conversely, any point not contained in at least $n-f$ measurements cannot be the correct value; if it were, then there would be more than f faulty sensors. So, the cover of all $(n-f)$-*cliques* must contain the correct value. (An $(n-f)$-*clique* corresponds to a value where at least $(n-f)$ sensor measurements intersect.)

We have one further constraint: any program written to deal with a single measurement assumes that the sensor delivers a region of some expected shape (e.g., rectangle, sphere, interval, etc.), so we require the cover to also have this same shape. This constraint allows us to improve a program based on a single (unreliable) abstract sensor by changing only the implementation of the sensor; the abstract sensor is replaced by several abstract sensors whose inputs are combined to produce a single reliable abstract sensor. The program can use the resulting reliable abstract sensor just as it originally used the single abstract sensor.

To summarize, we have the following goals for our reliable abstract sensor:

1. It should be guaranteed (assuming no more than f failures) to deliver a region containing the true physical value.

2. It should deliver a shape that is within the same class as the shapes delivered by the individual abstract sensors.

3. It should be accurate. In other words, assuming no more than f failures, it should deliver a region that is not significantly larger than a region that might be delivered by a single, correct abstract sensor.

4. It should be efficient to compute. A reliable abstract sensor is useless unless it can be computed in a reasonable amount of time.

It is useful to define $\mathcal{I}_{f,n}(S)$, the smallest region the satisfies goals 1 and 2. In other words, $\mathcal{I}_{f,n}(S)$ is the smallest figure of the correct shape that covers all $(n-f)$–cliques in S. For instance, if the individual sensors report intervals in one dimension then $\mathcal{I}_{f,n}(S)$ is the smallest interval that contains all the $(n-f)$–cliques. It is clear that the (unknown) true value $\overline{v}$ is a member of $\mathcal{I}_{f,n}(S)$ as long as no more than f measurements are faulty.

Figure 3 illustrates $\mathcal{I}_{f,n}(S)$ for measurements that are rectangles. The left-hand figure shows four measurements, and the right-hand figure shows the smallest rectangle that covers all 3–cliques of the measurements.

3 Linear Sensors

In [6], we show that for linear sensors - sensors that report 1D intervals - $\mathcal{I}_{f,n}(S)$ can be found efficiently and that for $f < \frac{n}{2}$, $\mathcal{I}_{f,n}(S)$ has reasonable size. The upper bounds on $|\mathcal{I}_{f,n}(S)|$ are stated in the following two theorems. We do not include the proofs in this paper, but the bounds are derived by considering *interval graphs* [4].

First, we need some notation. Define the functions $\mathbf{min}_i$ and $\mathbf{max}_i$ to be the i^{th} smallest and largest values of a set of n values respectively. Note that $\mathbf{min}_i$ is the same as $\mathbf{max}_{n-i+1}$. For example, if $S = \{13, 14, 15\}$ then $\mathbf{min}_3(S) = \mathbf{max}_1(S) = 15$.

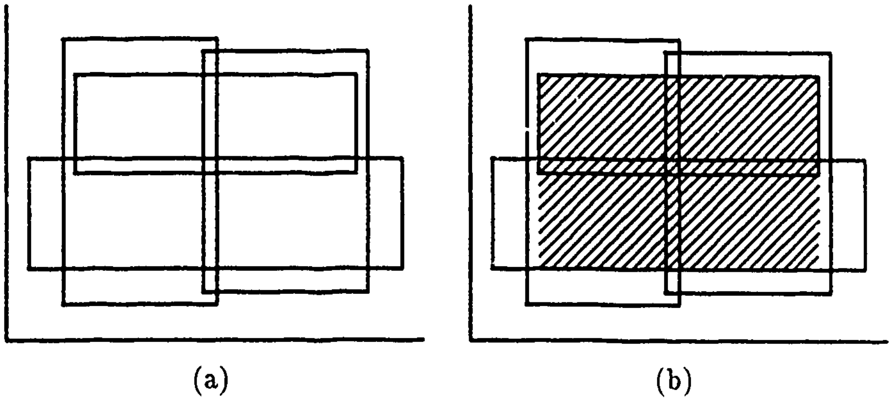

Figure 3: $\mathcal{I}_{1,4}(S)$ for Rectangular Measurements.

Theorem 1 *Let S be a set consisting of n intervals. If* $0 \leq f < \frac{n}{2}$ *then* $|\mathcal{I}_{f,n}(S)| \leq \min_{2f+1}\{|\overline{s}| : \overline{s} \in S\}$.

Thus, when $f < \frac{n}{2}$, the resulting reliable abstract sensor is as accurate as one of the original sensors, and the larger $n - f$ is (i.e., the more likely any one sensor reading is correct), the more accurate $\mathcal{I}_{f,n}(S)$ is.

$\mathcal{I}_{f,n}(S)$ can also be computed efficiently – in $O(n \log n)$ time – by sorting the endpoints of the n intervals and then moving through the endpoints in order, keeping track of the depth at each instant. $\mathcal{I}_{f,n}(S)$ is bounded by the smallest and largest points that are in $(n - f)$-cliques. Figure 4 illustrates this algorithm. The hatched areas denote the points that are in 3-cliques, and the lowest interval is $\mathcal{I}_{2,5}(S)$. Note that according to Theorem 1, the length of $\mathcal{I}_{2,5}(S)$ is bounded by the length of the longest interval in S, although in Figure 4, it happens to be shorter than the longest interval.

The second theorem states that there is no upper bound on the size when $f \geq \frac{n}{2}$.

Theorem 2 *Given a set $\{\ell_1, \ell_2, ..., \ell_n\}$ of n lengths and $\frac{n}{2} \leq f < n$, then for any length $\Lambda \geq \max\{\ell_1, \ell_2, ..., \ell_n\}$, there exists a set of n intervals $S = \{\overline{s}_1, \overline{s}_2, ..., \overline{s}_n\}$ where $\forall i : 1 \leq i \leq n : |\overline{s}_i| = \ell_i$ and $|\mathcal{I}_{f,n}(S)| = \Lambda$.*

It is easy to see that an equivalent of Theorem 2 holds for multidimensional sensors as well as linear ones. If over half the sensors have failed then $\mathcal{I}_{f,n}(S)$ may be arbitrarily large regardless of the dimension of the sensor's data.

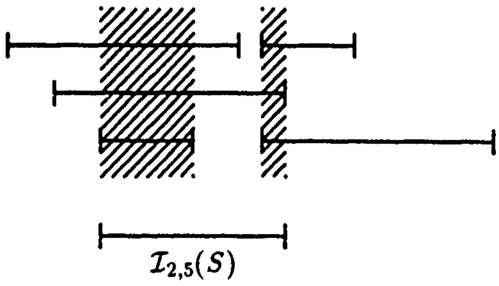

Figure 4: $\mathcal{I}_{2,5}(S)$ for Linear Measurements.

3.1 Multidimensional Sensors and Projection

The 1D results on intervals can be used directly to give results for multidimensional sensors. For a d–dimensional sensor, we project the region for sensor measurement s_i onto each of the d orthogonal axes. We now have d separate 1D problems. These problems can be solved individually and then recombined to produce a d–rectangle, which we call the *projection rectangle.*

There are several possible disadvantages to this approach:

1. Information may be lost. For example, the knowledge that a sensor's x–coordinate cannot possibly be correct can be used to discard the entire measurement.

2. A d–rectangle is not necessarily the desired shape. For example, our abstract sensor may be required to report a circle.

3. The size of the resulting sensor may be larger than necessary (for example, see Figure 5).

In fact, projection techniques are the method-of-choice in some situations (see Section 4), but these situations depend on the shapes involved and the relationship between f and n.

4 d–Rectangles

If s_i is constrained to be a d–dimensional rectangle, then another upper bound can be placed on the size of $\mathcal{I}_{f,n}(S)$.

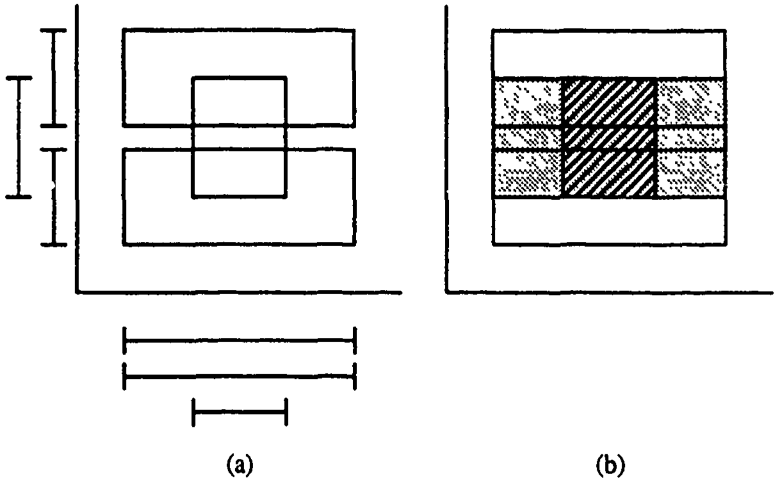

Figure 5: (a): Three rectangles and their projection onto the x and y axes. (b): $\mathcal{I}_{1,3}(S)$ is the crosshatched region and the projection rectangle is the gray region.

Theorem 3 *Let S be a set consisting of n d-dimensional rectangles. If $0 \leq f < \frac{n}{2d}$ then $|\mathcal{I}_{f,n}(S)| \leq \min_{2df+1}\{|\overline{s}| : \overline{s} \in S\}$.*

Proof. We use a counting argument to show that $\mathcal{I}_{f,n}(S)$ is contained in at least $(n - 2df)$ of the original d-rectangles. Assume $f < \frac{n}{2d}$. Choose $2d$ points, one from each of the $2d$ sides of $\mathcal{I}_{f,n}(S)$ where each chosen point is a member of an $(n - f)$-clique. These points must exist since if they did not, $\mathcal{I}_{f,n}(S)$ could be reduced in size. Call this set P. By definition of $(n - f)$-clique, each point p of P is contained in at least $(n-f)$ d-rectangles. Letting R_p represent the set of d-rectangles containing p, we have $n - f \leq |R_p|$ for each point $p \in P$. If we sum the number of rectangles containing each point, we get

$$2d(n-f) \leq \sum_{p\in P} |R_p| = \sum_{i=1}^{2d} i*|\{\text{rectangles containing exactly } i \text{ points of } P\}|.$$

The last sum can be broken into two pieces: the part due to d-rectangles that contain all the points of P and the part due to d-rectangles that contain fewer points. Let a be the number of d-rectangles that contain all $2d$ of the points in P; that is, the rectangles that contain $\mathcal{I}_{f,n}(S)$. The number of rectangles remaining is $n - a$. The part of the sum due to the d-rectangles that contain fewer points can be bounded by $(2d - 1)(n - a)$. We now have

$$2d(n - f) \leq 2da + (2d - 1)(n - a).$$

Solving for a, we get $a \geq n - 2df$; thus, $\mathcal{I}_{f,n}(S)$ is contained in at least $(n - 2df)$ d-rectangles and the bound of the theorem follows immediately. □

The bound on f given in the theorem is tight. Figure 3 shows a 2D example where $f = \frac{n}{2d}$ and $\mathcal{I}_{f,n}(S)$ is larger (in area) than any of the original rectangles. Similar examples can be built for any dimension d.

This theorem shows that the increased accuracy comes with a price: if it is desired that $|\mathcal{I}_{f,n}(S)|$ be at least as accurate as some measurement in S, then the amount of replication needed increases quickly (linearly) with d. For example, in order to tolerate a single failure for measurements that are $3D$ rectangles, a sensor must be replicated at least 7 times.

4.1 Algorithms for Rectangles

The 1D algorithm for intervals can be extended to handle rectangles. In 1D, we move from left to right across the intervals, keeping track of the covering depth. A similar *sweeping* idea works for 2D: we move a vertical sweep line from left to right across the rectangles, keeping track of the covering depth. Note that this depth can be different for different y-values, so depth information must be kept for each position along the vertical sweep-line. As the line enters or leaves a rectangle the depth information is updated. Using a naive implementation, this update takes $O(n)$ time, leading to an $O(n^2)$ time algorithm for computing $\mathcal{I}_{f,n}(S)$. Since the entire boundary of the $(n - f)$-cliques can be of complexity $\Omega(n^2)$, this might appear to be the best time-bound one can hope for. Note though, that the entire boundary is unnecessary; we need only determine the left, right, top, and bottom boundaries. This can be done efficiently by using Bentley's *segment tree* (see, for instance, [9]) to keep track of depth information along the vertical sweep-line. Thus, the entire computation for constructing $\mathcal{I}_{f,n}(S)$ can be done in $O(n \log n)$, including the initial sorting that must be done in preparation for both the sweep-line (sorting by x-coordinate) and the segment tree (sorting by y-coordinate).

Unfortunately, this technique does not generalize well to higher dimensions. The 2D version is fast because we can make use of the segment tree, a structure that allows efficient insertion and deletion of segments. But higher-dimensional analogs – allowing insertion and deletion of rectangles, for instance – are not correspondingly efficient. Thus, as the dimension increases the time bounds become prohibitively large.

There is however, an efficient algorithm that reports a d–rectangle (for any dimension d) that is almost as good as the minimal d–rectangle that we desire. This uses the projection technique described in Section 3.1, converting a d-dimensional problem into d 1-dimensional problems. The results of these separate 1D problems are combined to produce the *projection rectangle*, a d-rectangle that is guaranteed to be of reasonable size. The algorithm is based on the following theorem.

Theorem 4 *Let S be a set consisting of n d-dimensional rectangles. If $0 \leq f < \frac{n}{2d}$ then the size of the projection rectangle is $\leq \min_{2df+1}\{|\overline{s}| : \overline{s} \in S\}$.*

Proof. Each d-rectangle r is associated with exactly d intervals, one for each axis; these are the intervals found by projecting r onto the axes. Let I_r be the set of intervals associated in this way with d-rectangle r. For each axis, we now have a 1D problem with $f < \frac{n}{2d}$. By the proof of Theorem 3, the 1D $(n-f)$-cliques for each axis are contained in at least $n - 2f$ intervals. Let I be the set of all such intervals, at least $n-2f$ of them from each axis. If we sum the number of rectangles over all intervals, we get

$$d(n-2f) \leq |I| = \sum_{i=1}^{d} i * |\{r : |I_r \cap I| = i\}|.$$

The last sum can be broken into two pieces: the part due to rectangles that project onto a member of I for all axes, and the part due to other rectangles. Let a be the number of d-rectangles r for which $|I_r \cap I| = d$; that is, the d-rectangles that contain the projection rectangle. The number of rectangles remaining is $n - a$. The part of the sum due to these remaining rectangles can be bounded by $(d-1)(n-a)$. We now have

$$d(n-2f) \leq da + (d-1)(n-a).$$

Solving for a, we get $a \geq n - 2df$; thus the projection rectangle is contained in in at least $n - 2df$ d-rectangles and the bound in the theorem follows immediately. □

Note that the projection rectangle can be computed in $O(dn \log n)$ time and has exactly the same size bound as $\mathcal{I}_{f,n}(S)$. Thus, if our goal is simply to create an abstract sensor that is at least as accurate as some measurement in S, the projection rectangle is as good as $\mathcal{I}_{f,n}(S)$.

This theorem shows that, at least for rectangles, the projection rectangle can be used to define an reliable abstract sensor with all the desirable properties that we have specified. The projection rectangle is either the same size or somewhat larger than $\mathcal{I}_{f,n}(S)$, the optimal rectangle.

5 *d*–Circles

In this section, we show that circles are better than rectangles in the sense that the bound on the size of $\mathcal{I}_{f,n}(S)$ for circles grows more slowly than the corresponding bound for rectangles. We also show that circles are worse than rectangles in the sense that $\mathcal{I}_{f,n}(S)$ is more difficult to compute for circles than for rectangles.

If s_i is constrained to be a d-dimensional circle (e.g., a sphere in 3D) then the following upper bound can be placed on the size of $\mathcal{I}_{f,n}(S)$:

Theorem 5 *Let S be a set consisting of n d-circles. If $0 \leq f < \frac{n}{d+1}$ then $|\mathcal{I}_{f,n}(S)| \leq \min_{(d+1)f+1}\{|\bar{s}| : \bar{s} \in S\}$.*

Proof. We use a counting argument to show that $\mathcal{I}_{f,n}(S)$ is contained in at least $(n - (d+1)f)$ of the original d-circles. Assume $f < \frac{n}{d+1}$. Choose a set P, consisting of $d+1$ points such that each point is a member of an $(n-f)$-clique and the $d+1$ points *pin* the circle $\mathcal{I}_{f,n}(S)$. (A circle is *pinned* by a set of points if it is the smallest circle that includes that set of points.) These points must exist since if they did not, $\mathcal{I}_{f,n}(S)$ could be reduced in size. By definition of $(n-f)$-clique, each point p of P is contained in at least $(n-f)$ d-circles. Letting C_p represent the set of d-circles containing p, we have $n - f \leq |C_p|$ for each point $p \in P$. If we sum the number of circles containing each point, we get

$$(d+1)(n-f) \leq \sum_{p \in P} |C_p| = \sum_{i=1}^{d+1} i * |\{\text{circles containing exactly } i \text{ points of } P\}|.$$

The last sum can be broken into two pieces: the part due to d-circles that contain all the points of P and the part due to d-circles that contain fewer points. Let a be the number of d-circles that contain all $d+1$ of the points in P; that is, the circles that contain $\mathcal{I}_{f,n}(S)$. The number of circles remaining is $n - a$. The part of the sum due to the d–circles that contain fewer points can be bounded by $d(n - a)$. We now have

$$(d+1)(n-f) \leq (d+1)a + d(n-a).$$

Solving for a, we get $a \geq n - (d+1)f$; thus, $\mathcal{I}_{f,n}(S)$ is contained in at least $n-(d+1)f$ d–circles and the bound of the theorem follows immediately. □

This bound grows more slowly with d than does the bound of Theorem 3. For example, in order to tolerate a single failure for measurements that are spheres, a sensor must be replicated at least 4 times.

This theorem applies to sensors with a large variety of shapes - not just (simple) circles. Given a class of convex shapes in which the shapes are geometrically similar and share the same orientation, the shapes can be *pinned* by $d+1$ points where d is the dimension of the space. This property was the only circle property used in the proof of the theorem; thus, the same bounds hold for any such class of convex shapes.

Algorithms for d–circles are not as efficient as algorithms for d–rectangles. Even in 2D, it appears that to find the $(n - f)$–cliques, it is necessary to build the entire arrangement of n circles. Since n circles can have $\Omega(n^2)$ intersections, building the arrangement must take time $\Omega(n^2)$. (The incremental algorithm for building an arrangement of circles takes worst-case time $O(n\lambda_4(n))$ where λ_4 is an almost-linear function related to Davenport-Schinzel sequences [3]; using randomization, the arrangement can be built in expected time $O(m + n \log n)$ where m is the number of intersections [7].) Of course, we can replace each d–circle by a d–square that contains it and use the rectangle techniques, but this may produce an answer less accurate than desired.

6 Other Results

Improved results are possible if sensors are known to report d–rectangles that are all the same size and orientation. In this case, the projection technique can be used to create a reliable abstract sensor which reports a d–rectangle of the standard size in $O(dn \log n)$ time provided $f < \frac{n}{2}$. Note that for this case, the required relation between f and n is independent of d. This better bound occurs because for a single axis each projected rectangle (i.e., each interval) is exactly the same size. Since $f < \frac{n}{2}$, by Theorem 1 there is a single interval that contains all the $(n - f)$–cliques for an axis. When these containing intervals are recombined to create the projection

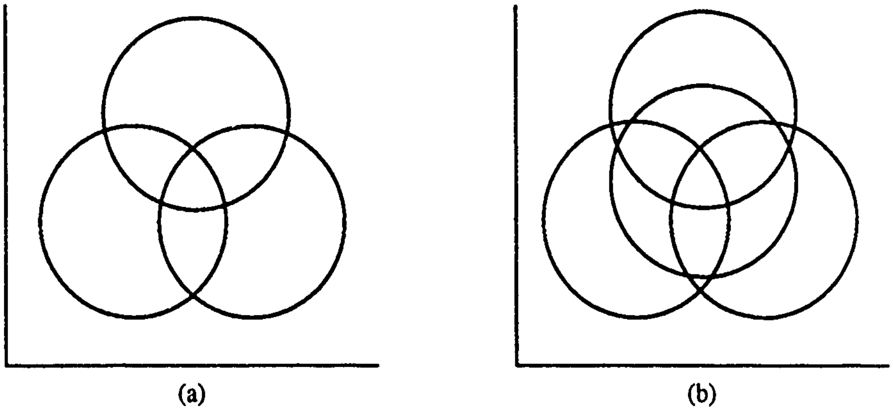

Figure 6: (a): Three unit circles. (b): Three unit circles with fourth unit circle in center. Note $\mathcal{I}_{1,3}(S)$ for the three original circles extends past the central unit circle.

rectangle we get a rectangle of the same size and orientation as the original rectangles. Note that he projection rectangle may not correspond to any of the original rectangles. In contrast, for identically sized circles, the smallest circle covering all of the $(n-f)$-cliques may be larger than the initial circles even when $f < \frac{n}{2}$. An example of this case is shown if Figure 6. Of course, the bound in Theorem 5 still applies; $|\mathcal{I}_{f,n}(S)|$ is bounded by the size of the initial circles when $f < \frac{n}{d+1}$.

Theorems 3 and 5 apply when measurements that are too inaccurate can be detected and removed in a preprocessing step. If this is not the case, then $\mathcal{I}_{f,n}(S)$ may be bounded by an abstract sensor that is too inaccurate. The following two theorems give bounds when abstract sensors may be undetectably inaccurate. Note that in this situation, a faulty sensor can contain the correct value.

Theorem 6 *Let S be a set consisting of n d-rectangles, and let C be the (unknown) subset of S that are correct. If $f < \frac{n}{2d+1}$ then $\mathcal{I}_{f,n}(S) \leq \mathbf{min}_{(2d-1)f+1}\{|s| : s \in C\}$.*

The proof of this theorem is simple: from Theorem 3,

$$|\mathcal{I}_{f,n}(S)| \leq \mathbf{max}_{n-2df}\{|s| : s \in S\}$$

For $|\mathcal{I}_{f,n}(S)|$ to be bounded by an accurate measurement, we must have $n - 2df > f$ and so $n > (2d+1)f$. The worst case is when f faulty measurements are the most inaccurate, so

$$|\mathcal{I}_{f,n}(S)| \leq \min_{(2d-1)f+1}\{|s| : s \in \mathcal{C}\}$$

□

A similar proof supports the following theorem:

Theorem 7 *Let S be a set consisting of n d-circles, and let $\mathcal{C}$ be the (unknown) subset of S that are correct. If $f < \frac{n}{d+2}$ then $\mathcal{I}_{f,n}(S) \leq \min_{df+1}\{|s| : s \in \mathcal{C}\}$.*

We have also looked at some fast approximation techniques. A grid of equal-sized buckets can be used to detect $(n-f)$-cliques, leading to a linear-time fault-tolerant averaging algorithm at the cost of some accuracy. This technique works for both d–rectangles and d–circles, but is more accurate for rectangles.

7 Summary

We have shown how several abstract sensors (that measure the same multidimensional physical data) can be combined to produce a *reliable abstract sensor*. This process can be done efficiently for d–rectanlges, reporting a region guaranteed to be of reasonable size, provided $f < \frac{n}{2d}$ where n is the number of sensors and f is the number of sensors that are faulty. For d–circles, a reliable abstract sensor region of reasonable size exists provided $f < \frac{n}{d+1}$, but determining this region is considerably less efficient. As mentioned above, the results on size bounds for circles actually hold for any class of convex shapes in which the shapes are geometrically similar and share the same orientation.

The following table summarizes our results:

geometry	n	complexity	comments
linear	$2f+1$	$O(n \log n)$	
rectangles	$4f+1$	$O(n \log n)$	
d–rectangles	$2df+1$	unacceptable	with $\mathcal{I}_{f,n}(S)$
d–rectangles	$2df+1$	$O(dn \log n)$	with projection technique
circles	$3f+1$	$O(n^2)$	randomized
d–circles	$(d+1)f+1$	unacceptable	
d–rectangles	$2f+1$	$O(dn \log n)$	uniform size

The results in this table assume that the goal is to produce a reliable abstract sensor whose size is no larger than that of a single individual abstract sensor. If the reliable abstract sensor is allowed to be somewhat larger, then many of the time bounds can be improved. For instance, d–circles can be approximated by d–squares in order to produce a less-accurate reliable abstract sensor in time $O(dn \log n)$ by using the projection technique.

Theorem 3 shows bounds on the size of a reliable abstract sensor for $f < \frac{n}{2d}$ and an analog of Theorem 2 shows that for $f \geq \frac{n}{2}$ the size of an abstract sensor is unbounded. For in-between values of f, $\frac{n}{2d} \leq f < \frac{n}{2}$, reliable abstract sensors are of bounded size, but such a sensor may report a d-rectangle significantly larger than any of the original d-rectangles.

References

[1] Flaviu Cristian, Houtan Aghili, and Ray Strong. Atomic broadcast: From simple message diffusion to Byzantine agreement. Technical Report RJ 5244 (54244), IBM Almaden Research Laboratory, July 1986.

[2] Wilbur L. Davenport and William L. Root. *An Introduction to the Theory of Random Signals and Noise.* IEEE Press, 1987.

[3] Herbert Edelsbrunner, Leonidas J. Guibas, Janos Pach, Richard Pollack, Raimund Seidel, and Micha Sharir. Arrangements of curves in the plane – topology, combinatorics, and algorithms. Technical Report UIUCDCS-R-88-1477, University of Illinois at Urbana-Champaign, December 1988.

[4] Martin C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, 1980.

[5] Leslie Lamport. Using time instead of timeout for fault-tolerant distributed systems. *ACM Transactions on Programming Languages and Systems*, 6(2):254–280, April 1984.

[6] Keith Marzullo. Tolerating failures of continuous–valued sensors. *ACM Transactions on Computer Systems*, to appear. Available as Technical Report TR 90-1156, Cornell University, September 1990.

[7] Ketan Mulmuley. A fast planar partition algorithm, II. In *Proceedings of the Fifth Annual Symposium on Computational Geometry*, pages 33–43. ACM Press, June 1989.

[8] Gil Neiger and Sam Toueg. Automatically increasing the fault-tolerance of distributed systems. In *Proceedings of the Eighth Symposium on Principles of Distributed Computing*, pages 248–262. ACM SIGPLAN/SIGOPS, August 1988.

[9] Franco P. Preparate and Michael I. Shamos. *Computational Geometry*. Springer-Verlag, 1985.

[10] Fred B. Schneider. The state machine approach: A tutorial. *Computing Surveys*, 22(3), September 1990.

[11] R. A. Short. The attainment of reliable digital systems through the use of redundancy: A survey. *IEEE Computer Group News*, 2:2–17, March 1968.

[12] John von Neumann. Probabilistic logics and the synthesis of reliable organisms from unreliable components. In C. E. Shannon and J. McMarthy, editors, *Automata Studies*, pages 43–98. Princeton University Press, 1956.